

Serial No. 09/559,224  
Reply to Office Action of July 15, 2005

### REMARKS

Responsive to the Office Action mailed July 15, 2005, Applicant has studied the Examiner's comments and the cited art. Claims 1-54 are currently pending; after entry of this Amendment, claims 55-88 remain pending. In view of the following remarks, Applicant respectfully submits that the application is in condition for allowance.

#### Amendments

Applicant has cancelled the previously pending claims 1-54 and provided new claims 55-88 to better claim the disclosed subject matter. New claims 55-88 do not introduce new matter and are supported by the Specification.

#### Claim Objections

Claim 37 is objected to because of an informality. Applicant has cancelled claim 37.

#### Claim Rejections Under 35 U.S.C. § 103

Claims 1-54 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Faustini et al., U.S. Patent No. 5,842,020, in view of Hanson, U.S. Patent No. 5,956,736. Applicant has cancelled claims 1-54 and provided new claims 55-88 and will address the rejections as they apply to the new claims.

Neither Faustini nor Hanson, alone or in combination, teach or suggest all of the elements of Applicant's claimed subject matter.

With respect to independent claim 55, the Office Action asserts that Faustini teaches "an inspector object configured to permit modification of at least one application object in the object oriented application, for communicating information pertaining to the attributes of the application object while the application object is deployed in the execution environment ('Dynamic editing is accomplished by providing each component that would have need of an editor with that capability as an integral part of the class template from which it is instantiated.')<sup>1</sup> The editor method of Faustini that is integral to the class template from which it is instantiated does not execute in the execution environment independently of the component, because it is integral to the component. In contrast, Applicant's claimed subject matter recites the application object executing independently of the inspector object.

Furthermore, Faustini does not recite the use of an inspector document associated with the inspector object, where the inspector document describes an attribute of the application object, as in Applicant's claimed subject matter. All of the attribute information of Faustini is

---

<sup>1</sup> Paper 20050406, p. 3.

Serial No. 09/559,224

Reply to Office Action of July 15, 2005

coded into the editor component method of Faustini. For example, the code for the class netpinEditor<sup>2</sup> provides code to edit a property of the I/O pins 904.<sup>3</sup>

In addition, as admitted by the Office Action, Faustini fails to teach or suggest a server that provides a user interface to the inspector object, as in Applicant's claimed subject matter.<sup>4</sup> Nor does Faustini teach or suggest a server that is configured to add the application objects, inspector objects, and inspector documents to the execution environment, or that configures the inspector object and inspector document for communication with the application object, or that selects inspector objects or inspector documents to use for modification of the application objects.

The Office Action, in responding the previously pending claims, attempts to fill that gap with Hanson, arguing that "A document server is inherently taught by Hanson, where the collection of objects is maintained."<sup>5</sup> Even if Hanson suggests a document server, which Applicant denies, Hanson fails to teach or suggest a server that configures an inspector object and an inspector document for communication with an application object, as in Applicant's claimed subject matter. Furthermore, Hanson's HTML editor is not an execution environment in which the application objects and the inspector objects both execute. The Office Action interprets the "Object Editor Window" of Hanson as an inspector.<sup>6</sup> Similarly, the Office Action understands the HTML file edited by the Object Editor Window as an application object. Even if Hanson's HTML editor is an execution environment in which the "Object Editor Window" executes, which Applicant denies, the HTML document edited by the HTML editor of Hanson does not also execute in the HTML editor, but executes in the context of a web browser and web server, as is well known in the art. Therefore, the HTML editor of Hanson would not be understood by one of skill in the art as an execution environment in which both application objects and inspector objects execute.

Therefore, even if there is a justification to combine Faustini and Hanson, which Applicant does not admit, neither Faustini nor Hanson, alone or in combination, teaches or suggests all of the elements of Applicant's claimed subject matter. For at least these reasons, Applicant respectfully submits new claim 55 is allowable.

Dependent claims 56-70 depend from allowable independent claim 55 and are therefore also allowable, for at least this reason.

Furthermore, with regards to claim 56, Faustini and Hanson fail to teach or suggest the ability to have more than one inspector object configurable to communicate with a single application object. Instead, the editor method of each component of Faustini is integral to the component being edited.

---

<sup>2</sup> Col. 152, line 33-Col. 153, line 6.

<sup>3</sup> Col. 151, lines 10-11.

<sup>4</sup> Paper 20050406, p. 4.

<sup>5</sup> Paper 20050406, p. 4.

<sup>6</sup> Paper 20050406, p. 4.

Serial No. 09/559,224  
Reply to Office Action of July 15, 2005

Similarly, the integral editor method of Faustini is only capable of editing attributes of the component of which it is an integral part. Therefore, Faustini fails to teach or suggest an inspector object capable of editing multiple application objects, as in claim 57. In addition, neither Faustini nor Hanson, alone or in combination, teaches or suggests a server to configure an inspector object and an inspector document for communication with multiple objects.

With respect to claim 61, even if the editor methods of Faustini are inspector objects as in Applicant's claimed subject matter, which Applicant denies, the editor method of Faustini does not communicate with a method of the component to modify the application object, where the method of the application object in turn actually performs the modification. Rather, the integral editor method of Faustini makes the modifications directly.

For these additional reasons, applicant respectfully submits that claims 56, 57, and 61 are allowable.

With respect to independent claim 71, as shown above, Faustini, even in combination with Hanson, fails to teach or suggest a plurality of inspector objects and inspector documents corresponding to the inspector objects that are selected and configured by a server for communication with the application object. Furthermore, the HTML editor of Hanson does not instantiate the HTML editor objects in the execution environment of the HTML file being edited, thus does not teach or suggest a server as an applicant's claimed subject matter. For at least these reasons, Applicant respectfully submits claim 71 is allowable.

Dependant claims 72-76 depend from independent claim 71 and are therefore are also allowable. For at least this reason, Applicant respectfully submits claims 72-76 are allowable.

In addition, neither Hanson nor Faustini alone or in combination teaches or suggests a library of objects from which the server selects application objects, inspector objects, and inspector documents as an Applicant's claim 72. For this additional reason, Applicant respectfully submits claim 72 is allowable.

With respect to independent claim 77, neither Faustini nor Hanson teaches or suggests selecting an inspector object and executing the inspector object in the execution environment independently of the application object. As the Office Action admits, Faustini's editor methods are integral to the component, thus are not selected, nor do they execute independently of the component. The HTML editor objects of Hanson do not execute in the execution environment of the HTML file being edited, even if the HTML file can be considered an application object that executes in an execution environment, which Applicant does not admit. In addition, neither Faustini nor Hanson teaches or suggests an inspector document as in Applicant's claimed subject matter. For at least these reasons, Applicant respectfully submits claim 77 is allowable.

Dependent claims 78-84 depend from allowable independent claim 75 and are therefore also allowable. For at least this reason, Applicant respectfully submits claims 76-82 are allowable.

Serial No. 09/559,224

Reply to Office Action of July 15, 2005

In addition, with respect to claim 83, neither Faustini nor Hanson teaches or suggests dynamically discovering attributes of an application object by the inspector object and creating an inspector document corresponding to the discovered attributes. For these additional reasons, Applicant respectfully submits claim 83 is allowable.

With respect to independent claim 85, neither Faustini nor Hanson teaches or suggests a server that can selectively deploy applications from a library of application objects and assemble those executing objects into an application, where the application objects have no predetermined relationship with each other. Furthermore, even if Hanson inherently teaches a server, as asserted by the Office Action, which Applicant does not admit, nothing in Hanson teaches or suggests that this implied server is configured to modify the attributes of executing application objects. For at least these reasons, Applicant respectfully submits claim 85 is allowable.

Dependent claims 86-88 depend from allowable claim 85 and are therefore also allowable. For at least this reason, Applicant respectfully submits claims 86-88 are allowable.

In addition, as shown above, neither Faustini nor Hanson teaches or suggests an inspector object that executes independently of the first and second application objects that is configured by the server for modification of attributes of the first application object. For this additional reason, Applicant respectfully submits that claim 86 is allowable.

Furthermore, neither Faustini nor Hanson, alone or in combination, teaches or suggests an inspector document, where the server discovers attributes of the first application object and configures the inspector document to describe a first attribute of the attributes of the first application object discovered by the server, where the inspector object uses the inspector document to selectively modify the first attribute. For this additional reason, Applicant respectfully submits that claim 87 is allowable.

Similarly, neither Faustini nor Hanson teaches or suggests an inspector document that is preconfigured to describe a first attribute of the first application object, where the inspector document is configured to use the inspector document to selectively modify the first attribute. For this additional reason, Applicant respectfully submits that claim 88 is allowable.

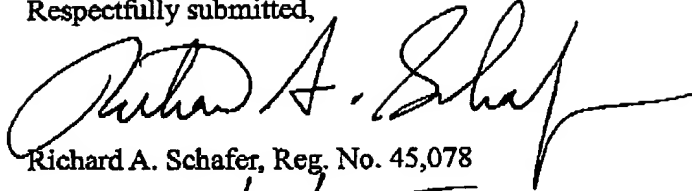
### **CONCLUSION**

Applicant respectfully submits that all issues and rejections have been adequately addressed, that all claims are allowable, and that the case should be advanced to issuance.

If the Examiner has any questions or wishes to discuss the claims, Applicant encourages the Examiner to call the undersigned at the telephone number indicated below.

Serial No. 09/559,224  
Reply to Office Action of July 15, 2005

Respectfully submitted,



Richard A. Schafer, Reg. No. 45,078

Date: 10/12/2005

AKIN GUMP STRAUSS HAUER & FELD LLP  
1111 Louisiana Street, 44<sup>th</sup> Floor  
Houston, Texas 77002-5200  
Telephone: (713) 220-5800  
Facsimile: (713) 236-0822